~~weight~~ is a drawing of a view screen that will be displayed in the personal computer in user subsystem 10, user station 170, and entry nodes 102, 104, 106, 108 and 110;

**FIG. 4** is a flow chart of the process for determining the cost of the other materials used to complete the container; and

**FIG. 5** is a flow chart showing how the post or carrier detects weight/rating/errors that are identified during normal article/mail piece acceptance processing.

***Page 8, lines 14 and 19:***

Subsystem **10** is shown as a node which includes a personal computer and printer for processing data and running certain software applications, a monitor for providing a human interface with the personal computer so as to view screens established by the application, and a keyboard for data entry. A modem link is implied that will allow access to interface **12**. Additional peripherals that are anticipated include a scanner for scanning barcodes and similar data~~, as well as a weighing scale for inputting weight should the system database **54** not contain a relevant weight for the article to be shipped~~.

***Page 8, line 29 and page 9, line 3:***

Database **54** is optionally linked to a weight and rate ~~rating~~ server **52** to form subsystem **50**. Database **54** comprises all universal process codes (UPC) database **54A**; item weight, volume and density databases **54B**; and all carrier rates database **54C**. Subsystem **50** is a remote server which can determine a rate for shipping a parcel in accordance with parameters established in the shipping application hosted by subsystem **10** and a weight returned from database **54**. Database **54** can be updated by

data entry from subsystem **10** or from periodic and/or random updates transmitted by postal or carrier server **28** and corrected or refined by database correction factors **32**.

***Page 9, lines 9 and 10:***

There are a number of commercially available databases that can serve as input to the trainable database **54**, thus reducing training time and expense. Those databases representing portions of the Universal Product Code (UPC) and the European Article Numbering (EAN) systems are the most advantageous because they contain product descriptions and characteristics but not the weight of the item. Currently, portions of these systems are commercially available, while a complete compilation is not available. ~~The trainable database contemplated herein would provide the quickest road to actually building a database that anticipates the entire UPC product base.~~

***Page 10, line 12:***

The first overhead digit of a UPC type A symbol is a number related to the type of product being described as follows:

0 = normal UPC code
1 = reserved
2 = products sold by weight
3 = health related products
4 = UPC code used without limits
5 = coupons
6 = normal UPC code
7 = normal UPC code
8 = reserved
9 = reserved

***Page 10, line 32 to page 11, line 3:***

Returning to FIG. 1A, subsystem **10** is shown as also linked to the postal or carrier stream through letter or parcel presorted metered permit, stamp, or PC-based meters/stamp payment and mailer finishing process~~activities~~ **22** which include the physical mailing of a letter or the shipping of a parcel. The entry of a letter or a parcel into the postal or carrier stream allows the carrier to apply an acceptance and delivery process **24** which produces samples of weight data **26** which are input to carrier server **28**. In turn, the carrier server **28** applies the weight database correction factors ~~to~~using the collected weight data so that ~~the weight~~ corrections can be input to the trainable database **54** through interface **34**, Internet link **36** and interface **38**.

***Page 12, line 1:***

A third embodiment of the present invention is shown in **FIG. 1C.** FIG. 1C is a block diagram of a data processing node, which is co-located with the database, and wherein the data processing node interfaces with a carrier server ~~in~~is performing rating routines prior to seeking carrier acceptance.

***Page 13, lines 22, 23:***

Turning to **FIG. 2A**, there is shown the method of the preferred embodiment of the present invention. The method begins at step 199 when a user enters data into screen 300 (FIG. 5). Then, in step **200**, ~~with~~ the initiation of a shipping or carrier management application begins (hereinafter referred to as a shipping application) in a data processing system. The application can be configured to access carrier data

representative of one carrier, or in the alternative, can be configured to select from among two or more carriers as based upon selection criteria selected by a system user. For example, such criteria can include: cost; desired date of delivery; available services; or, shipping mode.

***Page 14, first, second and third paragraphs:***

From step **200**, the method advances to step **202**~~02~~**204** ~~where data relative to the parcel to be shipped is entered into a weight determination routine of the shipping application. The data can be entered by a system operator utilizing a keyboard, by scanning a barcode related to the contents of the parcel, or by other similar data entry methods. When the data has been entered, the method advances to the query at step **204**~~, which asks if a weight has been entered for the ~~parcel~~article. If the response to the query is "YES," then the method advances to step **214** where the article data is applied directly to a rating routine for determining the rate to be charged for shipping the ~~parcel~~ article via the selected carrier. The method advances from step **214** ~~along a dual path~~ to step **216** where the total rate is returned for the container.~~and essentially simultaneously along path **A2** to re-enter the method flow at step **252** as is shown in **FIG. 2B**. At step **252**, parcel weight is returned for use by the rating routine~~. If, however, the response to ~~the query at~~ step **204** is "NO," then the method advances to step 201 to read the next data field from data fields **305, 307 and 309**. Then the program goes to ~~the query at~~ step **206**.

The query at step **206** asks if a UPC bar code is available for the parcel to be shipped. If the response to the query is "YES," then the method advances to step **212** where the UPC ~~barcode~~ value is compared to the UPC database to obtain an article

postal or carrier weight, and the item volume and density, if found. Now the program goes to step 205 to buffer the current article weight with the other weight parameters and the current total weight. Then the program goes to decision block 203 to check if the total weight has been determined. If step 203 has not determined the total weight, the program goes to step 201 to read the next data field. If step 203 determines the total weight, the program goes to step 213 to determine if other parameters, i.e., density and volume, are present in the record. If step 213 determines no new parameters are present, the program goes to step 214. If step 213 determines that new parameters are present, the program goes to step 350 (FIG. 4). Once the weight is obtained, the method advances directly to step **214**. However, if the response to the query at step **206** is "NO," then the method advances to the query at step **208**.

At step **208**, the method queries at to whether the ~~parcel~~article can be identified by ~~its characteristics~~a description of the article. If the response to the query is "YES," then the method advances to step **210** where the characteristics are input to the system and the corresponding UPC data and the ~~parcel~~container weight are determined. From step **210**, the method flow advances to step ~~214~~**205**. If the response to the query at step **208** is "NO," then the method advances along path **A1** to step **230** as is shown in FIG. 2B.

***Page 15, line 4:***

Returning to step **214**, the method applies the article data, including the weight obtained at step **252**, to a rating engine to determine the rate to be charged for shipping the parcel via the selected carrier. The rate is returned at step **216** and applied to the article at step **218** by indicating the rate on a corresponding carrier manifest, producing

a label (which generally lists the addressee as well) corresponding to the rate, or both. In step 219, the rate is applied for this container. The ~~parcel~~container is then prepared for shipping at step **220** and the routine for the parcel is concluded at step **222**.

***Page 15, third paragraph:***

Returning to step **230**, if the response to the query is "NO," then the method advances to the query at step **240**. At step **240**, the system queries as to whether or not a product description is available. If the response to the query is "NO," then the method advances to step 245. Step 245 requests that the user enter data that describes the product. Then the program advances to the query at step **246**. If~~,~~ however, the response to the query at step **240** is "YES," then the program goes to step 242 to compare elements of the product ~~comprising the parcel~~ description with elements in UPC database fields.~~are weighted utilizing any of a number of known weighting techniques. The weighted elements of the product are compared with elements of the database field at step **242** before advancing~~ Now the program advances to the query at step **244**.

***Page 16, line 1:***

Returning to step **236**, if the response to the query as to whether or not the product number is available is "NO," then the method advances to step **242** where the ~~weighted~~ elements of the product are compared with elements in the UPC~~of the~~ database field~~s~~ ~~at~~ before advancing to the query at step **244**. If the response to the query at step **236** is "YES," then the system conducts a search of the database by the manufacturer's product number before advancing to the query at step **244**.

***Page 16, lines 8-10:***

At step **244**, the system queries as to whether or not a match has been determined for the comparisons made of the manufacturer's name, product number, or description. If the response to the query is "NO," then the ~~method~~program advances to step 245 where the user is requested to enter the data. Then the program goes to ~~the query at~~ step **246**. However, if the response to the query at step **244** is "YES," then the method advances directly to step **252** and returns a weight to the cost/rating routine for use at step **214**.

***Page 16, second, third and fourth paragraphs:***

Turning to step **246**, the method queries as to whether or not the system user can enter (e.g., via keyboard entry or scanner entry) the weight directly to the routine. If the response to the query is "YES," then the data is entered into the entry fields of the routine at step **250**; otherwise, if the response to the query is "NO," then the program goes to step **247**. Step **247** sends to the user's display the following message: "Weight is not available. Take finished package to post/carrier for cost/rating". Then this routine goes to step **249** and then to step **222** to end. ~~system user can weigh the article utilizing a weighing scale at step **248**. The weight is then read, at step **250**, from the scale and entered either automatically or manually into the entry fields.~~ From step **250**, the method advances to step ~~**250**~~**252** where the weight is returned to the application in step **216** (FIG. 2A) for use in determining the cost/rate.

~~The weights database utilizes the weight entered at step **248**, together with the product description or product number to establish a new weights record which is entered into the weights database for subsequent comparisons; thus, the database is~~

~~trainable and potentially increases its ability to reduce the need for future weighing steps.~~

**FIG. 3** is a drawing of a view screen that will be displayed in the personal computer in user subsystem **10**, user station **170**, entry nodes **102, 104, 106, 108,** and **110**. The data entry screen **300** is made up of seven subscreens indicated as **301, 303, 305, 307, 309, 311, and 313**. Each subscreen in turn contains one or more required data elements that the user must enter to define the article/mail piece for cost/rating, or to provide the needed data to allow computation of the container and its contents weight.

The first sub data entry screen **301** is in turn subdivided into four user data entry fields. They are identified as **A, B, C,** and **D** in **301**. Each field allows the user to identify the chosen carrier **(A)**, the level of delivery service requested **(B)**, any other requested services **(C)**, and finally the destination information **(D)** that enables delivery.

The next data entry subscreen **303** provides a field **(E)** where the user could enter the actual final accurate shipping weight when it is already known to them. These circumstances are likely found in a manufacturer parcel-shipping site where standard boxes, packing and contents are combined in known arrangements.

The next three subscreens (**305, 307** and **309**) provide the numeric and text information that enables the present invention to operate.

Subscreen **305** contains data entry lines labeled F1, F2 and more. Each line has fields for both the UPC code assigned to the contents item, as printed on its label, or found in its description. The "F" lines are filled in with either the UPC or a description until all the items are accounted for.

The next subscreen **307** deals with the mail piece container. At least one "G" line must be selected, and either the UPC number entered or the description.

The next subscreen **309** deals with the packing and tape used to form the mail piece/container. At least one "H" line must be selected, and both the UPC number entered and a description of what was consumed.

Subscreens **311** and **313** are not for data entry. These subscreens inform the user about the status of the "mail-ability" in ("I") of the mail piece, and the current cost in (J).

FIG. 4 is a flow chart of the process for determining the cost of the other materials used to complete the container. The program begins in step **350** when step **213** (FIG. 2A) detects other parameters.

Step **350** reads the input records from subscreens **310-309** (FIG. 3). Then the program goes to decision step **352**. Step **352** determines whether or not other parameter data entries are present in the data record. If other parameters are not present, the program goes to step **214** (FIG. 2A). If other parameters are present in step **352**, the program goes to step **354** to read or compute values for the current postal weight; container volume; sum of the contents items and packing material density. In step **356**, the program subtracts all contents volumes found from the container volume. Then, in step **358** the program multiplies the computed volume difference by the given packing material density. In step **360**, the program adds the computed postal weight for the packing used to the current postal weight. In step **362**, the file is returned to step **214** (FIG. 2A)

FIG. 5 is a flow chart showing how the post or carrier detects weight/rating/cost/errors that are identified during normal article/mail piece acceptance processing. Prior to initiation of the corrective process, described in FIG. 5, a temporary article weight error file **371** is produced by a sortation process **370.** Sortation process **370** out sorts all verified mail piece records that exceed post/carrier established acceptance value for a user-produced weighing error. The user-produced weighing error is usually 3-5% of the total weight.

The corrective processing starts at step **373** when the first record stored in step **371** is read in. Next, at step **375**, the mail piece unique number that the system issued during the user cost/rating process, is read in from subscreen **301** data field (FIG. 3) and is retrieved from the archived user record 300.

Next the process moves to step **377** to establish if a user established "postal weight" was used. If this is the case (yes), the process moves to step **379** where it computes the value of the error, and then notifies the user and bills the user's account.

Next, at step **380**, the process checks to see if there are any more records to process in step **371**. If there are none to process, it moves to step **381** to clear the processed records in step **371** and then ends.

If there are additional files to process in step **371**, the process gets the next record at step **392** and moves back to step **373**. Returning to step **377**, if a user supplied postal weight was not entered by the user, the process moves to step **383** where it verifies that the weights entered by the user interactive process matches the current UPC-based weights. If a mismatch is found, the process moves to step **379** where it follows the flow already discussed. If all the component weights are found to

match those currently in the UPC database **54b** (FIG. 1A), the process moves to step **385** to locate other mail pieces that contain the same item.

The process next moves to query the item manufacture's database over the Internet to verify the weight. It then moves to step **389** to produce a correction if needed. Then the process goes to step **390** to add the correction to a Weight Corrections database update file 391. Then the process moves to step **380**. Then the system database **32** (FIG. 1A) is updated from time to time as needed using step 391.

***Page 17 through page 18, first paragraph: delete in their entirety***

~~The method flow begins with a comparison input initiated at step **240**. The input initiates, at step **260**, a comparison routine. The routine begins at step **262** where the product description is parsed into matching fields before advancing to step **264** where each of the fields is weighted in accordance with its ability to define or direct a search to a product or article in terms of its actual weight. From step **264**, the method advances to step **266** where an address is assigned to each field and a fixed value alphanumeric is added that corresponds to a product description in the weights database. At step **268**, each of the alphanumerics is matched against the most appropriate table (based on the alphanumerics) in the weights database to determine whether a match is to be returned.~~

~~The method queries at step **270** as to whether or not a match is returned. If the response to the query is "YES," then the weight of that alphanumeric is added to a weights tally field. If the response to the query is "NO," then the method advances to the query at step **274** which asks if there are any more matches. If the response to the query is "YES," then the method advances to step **276** where the method goes to the next alphanumeric to be matched. The method then goes to the next alphanumeric and~~

~~re-enters the flow at step **270**. However, if the response to the query at step **274** is "NO," then the method advances to step **278** and determines whether or not the total weight recorded in the weight tally field is greater than, or equal to, a pre-determined threshold weight.~~

~~If the weight in the weight tally field is not greater than or equal to the threshold weight, then the method advances to step **280**, determines that no match has been found at step **280** and then returns at step **282** to step **244** as is shown in FIG. 2B. If it is determined that the weight in the weight tally field is greater than or equal to the threshold weight, then the method will advance from step **278** to step **284**. The method determines at step **284** that a match has been determined and then returns at step **286** to step **252** as is shown in FIG. 2B.~~

***Page 18, first paragraph:***

~~While certain embodiments have been described above in terms of the system within which the subject method may be employed, the invention is not limited to such a context. The system and its embodiments shown in FIG. 1A, FIG. 1B, FIG. 1C, and FIG. 1D are examples of configurations for the invention, and the system elements are intended merely to exemplify the types of peripherals and combinations that are contemplated by the invention.~~

Substitute new **FIG. 1A** for old FIG. 1A;

Substitute new **FIG. 2A** for old FIG. 2A;

Substitute new **FIG. 3** for old FIG. 3; and

Add new **FIG. 4** and **FIG. 5.**